

### **Statement of Relevancy for JP 2003-174565**

This document was cited in an Office Action in Japanese Patent Application No. 2007-518086, which corresponds to International Patent Application No. PCT/US2005/019256, which corresponds to U.S. Patent Application No. 10/875,678, filed on June 25, 2004.

## IMAGE PROCESSOR, IMAGE PROCESSING METHOD, PROGRAM AND STORAGE MEDIUM

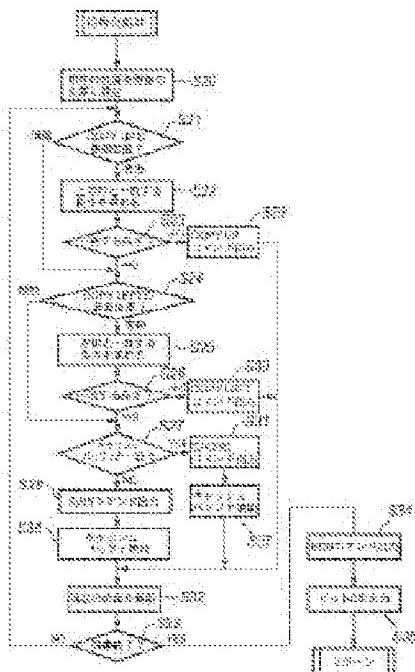
**Patent number:** JP2003174565 (A)  
**Publication date:** 2003-06-20  
**Inventor(s):** SAKAMOTO YOICHI +  
**Applicant(s):** CANON KK +  
**Classification:**  
- international: H03M7/40; H04N1/41; (IPC1-7): H03M7/40; H04N1/41  
- european:  
**Application number:** JP20010374340 20011207  
**Priority number(s):** JP20010374340 20011207

### Also published as:

JP3970007 (B2)

### Abstract of JP 2003174565 (A)

**PROBLEM TO BE SOLVED:** To provide high compressibility by using redundancy in the case of matching a previous image data string and to suppress decline of the compressibility as much as possible even in the case that a part where an image data string matches is small. ; **SOLUTION:** When encoding to a COPY UP command is possible, the COPY UP command and a code indicating the number of bytes following it are outputted (S29). When the encoding to a COPY LEFT command is possible, the COPY LEFT command and the code indicating the number of the bytes following it are outputted (S30). When the encoding to a CACHE command is possible, the CACHE command and data indicating a position of a cache buffer that a cache hits are outputted (S31). In the case that the byte at a present position is not registered in the cache buffer, a RAW command and raw data of 8 bits equal to the byte at the present position following it are outputted (S28). ; **COPYRIGHT:** (C)2003,JPO



Data supplied from the **espacenet** database — Worldwide

(19)日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-174565

(P2003-174565A)

(43)公開日 平成15年6月20日 (2003.6.20)

(51)Int.Cl.<sup>7</sup>H 04 N 1/41  
H 03 M 7/40

識別記号

F I

H 04 N 1/41  
H 03 M 7/40テマコト<sup>\*</sup>(参考)C 5 C 0 7 8  
5 J 0 6 4

審査請求 未請求 請求項の数21 O.L. (全 15 頁)

(21)出願番号 特願2001-374340(P2001-374340)

(22)出願日 平成13年12月7日 (2001.12.7)

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 坂本 陽一

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(74)代理人 100076428

弁理士 大塚 康徳 (外3名)

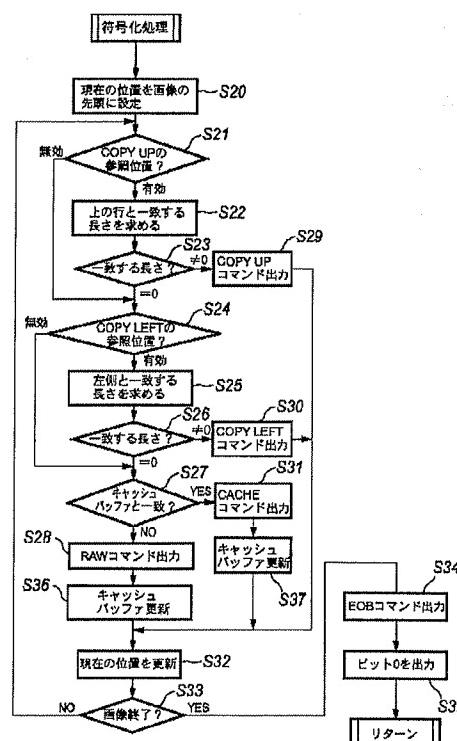
Fターム(参考) 5C078 AA09 BA22 CA01 DA01 DB05  
5J064 AA01 BA09 BC01 BC02 BC05  
BC14 BC29 BD07

(54)【発明の名称】 画像処理装置、画像処理方法、プログラム、及び記憶媒体

(57)【要約】

【課題】 以前の画像データ列と一致する場合にはその冗長性を利用して高い圧縮率を得るとともに、画像データ列が一致する部分が少ない場合においても、圧縮率の低下を極力抑えること。

【解決手段】 COPY UPコマンドに符号化できる場合、COPY UPコマンドとそれに後続するバイト数を示す符号を出力する (S 29)。COPY LEFTコマンドに符号化できる場合、COPY LEFTコマンドとそれに後続するバイト数を示す符号を出力する (S 30)。CACHEコマンドに符号化できる場合、CACHEコマンドとそれに後続する、キャッシュがヒットしたキャッシュバッファの位置を示すデータを出力する (S 31)。現在の位置のバイトがキャッシュバッファに登録されていない場合、RAWコマンドとそれに後続する現在の位置のバイトに等しい8ビットの生データを出力する (S 28)。



## 【特許請求の範囲】

【請求項1】 画像を圧縮する画像処理装置であって、画像に対してディザマトリックスを用いて色変換処理を行い、色毎の画像データを生成する色変換手段と、色毎の画像データに対して、圧縮対象の画素データ列と、当該画素データ列と所定の位置関係にある画素データ列とを比較し、一致する画像データ列の長さを示すコマンドと共に、当該比較内容を示すコマンドを出力し、色毎の画像データの符号化データを生成するコマンド出力手段とを備えることを特徴とする画像処理装置。

【請求項2】 画像を圧縮する画像処理装置であって、画像に対してディザマトリックスを用いて色変換処理を行い、色毎の画像データを生成する色変換手段と、前記色毎の画像データを圧縮し、符号化する符号化手段と、

前記符号化手段による符号化データを出力する出力手段とを備え、

更に前記符号化手段は、

圧縮対象の列の画素データ列と、当該列と所定の位置関係にある列の画素データ列とを比較して、一致する画素データ列の長さを求め、当該一致する画素データ列として、前記比較内容を示すコマンドと共に、前記長さを示すコマンドを出力する第1のコマンド出力手段と、前記長さが0の場合、圧縮対象の列の画素データ列と、当該列において所定の位置関係にある画素データ列とを比較して、一致する画素データ列の長さを求め、当該一致する画素データ列として、前記比較内容を示すコマンドと共に、前記長さを示すコマンドを出力する第2のコマンド出力手段と、

前記第1のコマンド出力手段により求めた長さ、前記第2のコマンド出力手段により求めた長さが共に0の場合、圧縮対象の列の画素データ列の一部もしくは全部を格納する格納手段と、

前記第1のコマンド出力手段により求めた長さ、前記第2のコマンド出力手段により求めた長さが共に0の場合、圧縮対象の画素データと前記格納手段に格納された画素データ列とを比較し、前記圧縮対象の画素データとして、一致する画素データの前記格納手段における位置を示すコマンドと共に、前記比較内容を示すコマンドを出力する第3のコマンド出力手段と、

前記第1のコマンド出力手段により求めた長さ、前記第2のコマンド出力手段により求めた長さが共に0の場合、且つ、前記第3のコマンド出力手段による比較の結果、圧縮対象の画素データと一致する画素データが前記格納手段に存在しないと判断された場合、圧縮対象の画素データを示すコマンドを出力する第4のコマンド出力手段とを備え、

前記第1乃至4のコマンド出力手段の一部もしくは全部によるコマンドを含む符号化データを生成することを特徴とする画像処理装置。

【請求項3】 前記色変換手段は、画像に対してディザマトリックスを用いることで、赤、緑、青を用いた画像をシアン、マゼンタ、黄、黒を用いた画像に色変換し、前記符号化手段は、シアン、マゼンタ、黄、黒の各色の画像データに対して符号化を行うことを特徴とする請求項2に記載の画像処理装置。

【請求項4】 前記第3のコマンド出力手段は、前記第1のコマンド出力手段により求めた長さ、前記第2のコマンド出力手段により求めた長さが共に0の場合、圧縮対象の画素データと前記格納手段に格納された画素データ列とを比較し、前記圧縮対象の画素データとして、一致する画素データの前記格納手段における位置を示すコマンドと共に、前記比較を示すコマンドを出力し、更に、前記一致する画素データを前記格納手段に格納することを特徴とする請求項2に記載の画像処理装置。

【請求項5】 前記第4のコマンド出力手段は、前記第1のコマンド出力手段により求めた長さ、前記第2のコマンド出力手段により求めた長さが共に0の場合、且つ、前記第3のコマンド出力手段による比較の結果、圧縮対象の画素データと一致する画素データが前記格納手段に存在しないと判断された場合、圧縮対象の画素データを示すコマンドを出力し、

更に、前記圧縮対象の画素データを前記格納手段に格納することを特徴とする請求項2に記載の画像処理装置。

【請求項6】 前記格納手段はキャッシュバッファであって、前記第3のコマンド出力手段は、圧縮対象の画素データとして、圧縮対象の画素データに対してキャッシュがヒットした前記キャッシュバッファの位置を示すコマンドと共に、前記比較内容を示すコマンドを出力することを特徴とする請求項2乃至5のいずれか1項に記載の画像処理装置。

【請求項7】 前記出力手段は、出力する符号化データの総ビット数が8の倍数となるように0を補填することを特徴とする請求項2に記載の画像処理装置。

【請求項8】 前記所定の位置関係は、前記ディザマトリックスの周期に応じたものであることを特徴とする請求項2に記載の画像処理装置。

【請求項9】 請求項2乃至8のいずれか1項に記載の画像処理装置による符号化データを復号する画像処理装置であって、

復号した画素データ列を格納する第1の格納手段と、復号した画素データ列を所定個数分シフトして格納する第2の格納手段と、

前記第4のコマンド出力手段により出力されたコマンドが表す画素データを格納する第3の格納手段と、前記第1乃至4のコマンド出力手段から出力されたコマンドの内容を特定し、特定した内容に応じて前記第1の格納手段、もしくは第2の格納手段、もしくは第3の格納手段に格納された画素データを用いて画像を復元する復元手段とを備えることを特徴とする画像処理装置。

【請求項10】 前記第1の格納手段はラインバッファであって、前記第1のコマンド出力手段により比較される夫々の画素データ列の間隔分のサイズを有するリングメモリであることを特徴とする請求項9に記載の画像処理装置。

【請求項11】 前記第2の格納手段はシフトレジスターとセレクタにより構成されており、復号した画素データ列を、前記第2のコマンド出力手段により比較される夫々の画素データの間隔分シフトすることを特徴とする請求項9に記載の画像処理装置。

【請求項12】 前記第3の格納手段はキャッシュメモリであることを特徴とする請求項9に記載の画像処理装置。

【請求項13】 前記復号手段は、前記第1のコマンド出力手段により出力されたコマンドの場合、当該コマンドによって特定される前記第1の格納手段に格納された画素データ列を読み出して、前記第1の格納手段に格納された画素データ列の最後尾に追加して格納すると共に、前記第2の格納手段の先頭に格納することを特徴とする請求項9乃至12のいずれか1項に記載の画像処理装置。

【請求項14】 前記復号手段は、前記第2のコマンド出力手段により出力されたコマンドの場合、当該コマンドによって特定される個数分の画素データ列を前記第2の格納手段から読み出して、前記第1の格納手段に格納された画素データ列の最後尾に追加して格納すると共に、前記第2の格納手段の先頭に格納することを特徴とする請求項9乃至12のいずれか1項に記載の画像処理装置。

【請求項15】 前記復号手段は、前記第3のコマンド出力手段により出力されたコマンドの場合、当該コマンドによって特定される前記第3の格納手段に格納された画素データを読み出して、前記第1の格納手段に格納された画素データ列の最後尾に追加して格納すると共に、前記第3の格納手段の先頭に格納することを特徴とする請求項9乃至12のいずれか1項に記載の画像処理装置。

【請求項16】 前記復号手段は、前記第4のコマンド出力手段によって出力されたコマンドの場合、当該コマンドが表す画素データを前記第1の格納手段に格納された画素データ列の最後尾に追加して格納すると共に、前記第3の格納手段の先頭に格納することを特徴とする請求項9乃至12のいずれか1項に記載の画像処理装置。

【請求項17】 画像を圧縮する画像処理方法であつて、

画像に対してディザマトリックスを用いて色変換処理を行い、色毎の画像データを生成する色変換工程と、色毎の画像データに対して、圧縮対象の画素データ列と、当該画素データ列と所定の位置関係にある画素データ列とを比較し、一致する画像データ列の長さを示すコ

マンドと共に、当該比較内容を示すコマンドを出し、色毎の画像データの符号化データを生成するコマンド出力工程とを備えることを特徴とする画像処理方法。

【請求項18】 画像を圧縮する画像処理方法であつて、

画像に対してディザマトリックスを用いて色変換処理を行い、色毎の画像データを生成する色変換工程と、前記色毎の画像データを圧縮し、符号化する符号化工程と、

10 前記符号化工程による符号化データを出力する出力工程とを備え、

更に前記符号化工程では、

圧縮対象の列の画素データ列と、当該列と所定の位置関係にある列の画素データ列とを比較して、一致する画素データ列の長さを求め、当該一致する画素データ列として、前記比較内容を示すコマンドと共に、前記長さを示すコマンドを出力する第1のコマンド出力工程と、前記長さが0の場合、圧縮対象の列の画素データ列と、当該列において所定の位置関係にある画素データ列とを比較して、一致する画素データ列の長さを求め、当該一致する画素データ列として、前記比較内容を示すコマンドと共に、前記長さを示すコマンドを出力する第2のコマンド出力工程と、

前記第1のコマンド出力工程で求めた長さ、前記第2のコマンド出力工程で求めた長さが共に0の場合、圧縮対象の画素データ列の一部もしくは全部を所定の格納手段に格納する格納工程と、

前記第1のコマンド出力工程で求めた長さ、前記第2のコマンド出力工程で求めた長さが共に0の場合、圧縮対象の画素データと前記格納工程で格納された画素データ列とを比較し、前記圧縮対象の画素データとして、一致する画素データの前記格納工程で格納した位置を示すコマンドと共に、前記比較内容を示すコマンドを出力する第3のコマンド出力工程と、

前記第1のコマンド出力工程で求めた長さ、前記第2のコマンド出力工程で求めた長さが共に0の場合、且つ、前記第3のコマンド出力工程での比較の結果、圧縮対象の画素データと一致する画素データが前記格納工程で格納していないと判断された場合、圧縮対象の画素データを示すコマンドを出力する第4のコマンド出力工程とを備え、

前記第1乃至4のコマンド出力手段の一部もしくは全部によるコマンドを含む符号化データを生成することを特徴とする画像処理方法。

【請求項19】 請求項18に記載の画像処理方法による符号化データを復号する画像処理装置であつて、

復号した画素データ列を第1の格納手段に格納する第1の格納工程と、

復号した画素データ列を所定個数分シフトして第2の格納手段に格納する第2の格納工程と、

前記第4のコマンド出力工程で出力されたコマンドが表す画素データを第3の格納手段に格納する第3の格納工程と、

前記第1乃至4のコマンド出力工程で出力されたコマンドの内容を特定し、特定した内容に応じて前記第1の格納工程、もしくは第2の格納工程、もしくは第3の格納工程で格納された画素データを用いて画像を復元する復元工程とを備えることを特徴とする画像処理方法。

【請求項20】請求項17乃至19のいずれか1項に記載の画像処理方法を実行するプログラム。

【請求項21】請求項20に記載のプログラムを格納し、コンピュータが読み取り可能な記憶媒体。

#### 【発明の詳細な説明】

##### 【0001】

【発明の属する技術分野】本発明は、画像を圧縮する、又はその符号を復号する画像処理装置、画像処理方法、プログラム、及び記憶媒体に関するものである。

##### 【0002】

【従来の技術】データを圧縮する方法として以前のデータ系列と一致する場合には、一致するデータの長さを符号化し、一致しない場合にはデータ自体を符号化するという方法がある。

【0003】例えば、LZ77圧縮法では、所定の大きな移動窓内の任意の位置のデータ系列と一致する場合には、一致するデータ系列の位置および一致する長さを符号化し、一致しない場合にはデータ自体を符号化する。

【0004】また画像データを圧縮する場合には、あらかじめ定められた1箇所または複数の箇所の位置、例えば符号化しようとしているデータの上あるいは左の位置のデータ系列と一致する場合には一致する長さを符号化し、一致しない場合にはデータ自体を符号化するという方法がある。

【0005】一方、特開平6-242924号公報に示されるように、最近出現したデータをキャッシュバッファに記憶し、キャッシュがヒットした場合には、一致したデータが記憶されているインデックスを符号化することで、データ自体を符号化する場合に較べて短い符号に符号化するという方法がある。

##### 【0006】

【発明が解決しようとする課題】しかしながら上述の方法では、データ系列が一致する部分が少ない場合にデータ自体を符号化した符号が多くなり、圧縮率が著しく低下するという欠点がある。

【0007】一方、特開平6-242924号公報に開示されている方法では、以前のデータ系列と一致する場合であってもその冗長性を圧縮に利用することができず、高い圧縮率を得ることが困難であった。

【0008】本発明は以上の問題に鑑みて成されたものであり、以前の画像データ列と一致する場合にはその冗

長性を利用して高い圧縮率を得るとともに、画像データ列が一致する部分が少ない場合においても、圧縮率の低下を極力抑えることを目的とする。

#### 【0009】

【課題を解決するための手段】本発明の目的を達成するために、例えば本発明の画像処理装置は以下の構成を備える。

【0010】すなわち、画像を圧縮する画像処理装置であって、画像に対してディザマトリックスを用いて色変換処理を行い、色毎の画像データを生成する色変換手段と、色毎の画像データに対して、圧縮対象の画素データ列と、当該画素データ列と所定の位置関係にある画素データ列とを比較し、一致する画像データ列の長さを示すコマンドと共に、当該比較内容を示すコマンドを出し、色毎の画像データの符号化データを生成するコマンド出力手段とを備える。

【0011】本発明の目的を達成するために、例えば本発明の画像処理装置は以下の構成を備える。

【0012】すなわち、画像を圧縮する画像処理装置であって、画像に対してディザマトリックスを用いて色変換処理を行い、色毎の画像データを生成する色変換手段と、前記色毎の画像データを圧縮し、符号化する符号化手段と、前記符号化手段による符号化データを出力する出力手段とを備え、更に前記符号化手段は、圧縮対象の列の画素データ列と、当該列と所定の位置関係にある列の画素データ列とを比較して、一致する画素データ列の長さを求め、当該一致する画素データ列として、前記比較内容を示すコマンドと共に、前記長さを示すコマンドを出力する第1のコマンド出力手段と、前記長さが0の場合、圧縮対象の列の画素データ列と、当該列において所定の位置関係にある画素データ列とを比較して、一致する画素データ列の長さを求め、当該一致する画素データ列として、前記比較内容を示すコマンドと共に、前記長さを示すコマンドを出力する第2のコマンド出力手段と、前記第1のコマンド出力手段により求めた長さ、前記第2のコマンド出力手段により求めた長さが共に0の場合、圧縮対象の列の画素データ列の一部もしくは全部を格納する格納手段と、前記第1のコマンド出力手段により求めた長さ、前記第2のコマンド出力手段により求めた長さが共に0の場合、圧縮対象の画素データと前記格納手段に格納された画素データ列とを比較し、前記圧縮対象の画素データとして、一致する画素データの前記格納手段における位置を示すコマンドと共に、前記比較内容を示すコマンドを出力する第3のコマンド出力手段と、前記第1のコマンド出力手段により求めた長さ、前記第2のコマンド出力手段により求めた長さが共に0の場合、且つ、前記第3のコマンド出力手段による比較の結果、圧縮対象の画素データと一致する画素データが前記格納手段に存在しないと判断された場合、圧縮対象の画素データを示すコマンドを出力する第4のコマンド出

力手段とを備え、前記第1乃至4のコマンド出力手段の一部もしくは全部によるコマンドを含む符号化データを生成する。

【0013】本発明の目的を達成するために、例えば本発明の画像処理装置は以下の構成を備える。

【0014】すなわち、上記画像処理装置による符号化データを復号する画像処理装置であって、復号した画素データ列を格納する第1の格納手段と、復号した画素データ列を所定個数分シフトして格納する第2の格納手段と、前記第4のコマンド出力手段により出力されたコマンドが表す画素データを格納する第3の格納手段と、前記第1乃至4のコマンド出力手段から出力されたコマンドの内容を特定し、特定した内容に応じて前記第1の格納手段、もしくは第2の格納手段、もしくは第3の格納手段に格納された画素データを用いて画像を復元する復元手段とを備える。

【0015】

【発明の実施の形態】以下添付図面を参照して、本発明を好適な実施形態に従って詳細に説明する。

【0016】【第1の実施形態】図11に本実施形態の画像処理装置の基本構成を示す。本実施形態では画像処理装置として一般的なパーソナルコンピュータやワークステーションを用いる。

【0017】1101はCPUで、RAM1102やROM1103に格納されたプログラムやデータを用いて本装置全体の制御を行うと共に、後述の画像圧縮処理も行う。1102はRAMで、外部記憶装置1104や記憶媒体ドライブ1109からロードされたプログラムやデータ等を一時的に記憶するエリアを備えると共に、CPU1101が各種の処理を実行する際に用いるワークエリアも備える。1103はROMで、本装置全体の制御プログラム（例えばブートプログラム）や制御データ（例えば本装置の設定データ）を格納する。1104はハードディスクなどの外部記憶装置で、記憶媒体ドライブ1109からインストールされたプログラムやデータなどを保存する。また、上述のワークエリアのサイズがRAM1102内に設けられなくなった場合に、不足分をファイルとして提供することもできる。1105、1106は夫々キーボード、マウスで、夫々ポインティングデバイスとして用いられ、各種の指示を本装置に入力することができる。1107は表示装置で、CRTや液晶画面などにより構成されており、画像や文字などを表示することができる。1108が画像入力装置で、デジタルカメラやスキャナなどにより構成されており、撮像や取り込みなどの操作により画像をデジタルデータとしてRAM1102や外部記憶装置1104に入力することができる。

【0018】1109は記憶媒体ドライブで、CD-R ROMやDVD-ROMなどの記憶媒体からプログラムやデータなどを読み込み、読み込んだプログラムやデータ

をRAM1102や外部記憶装置1104等に出力する。1110はI/F（インターフェース）で、インターネットやLANなどを介して外部の機器から圧縮処理対象の画像を受信したり、逆に圧縮した画像を外部の機器に対して送信する等、外部の機器とのデータの送受信を行う際のI/Fとして機能する。1111はプリンタで、圧縮された画像を伸長して紙などの記録媒体に印刷を行う画像伸長装置として機能する。1112は上述の各部を繋ぐバスである。

【0019】図1は画像を印刷する際に用いるソフトウェア群とプリンタとの関係を示す概念図である。外部記憶装置1104内にはオペレーティングシステム2（以下、OS）、アプリケーション3、プリンタドライバ4、ポートドライバ6が格納されている。

【0020】OS2は図1に示した各部（プリンタ1111は除く）、及びアプリケーション3、プリンタドライバ4、ポートドライバ5などのソフトウェアを管理する。アプリケーション3は、例えばワードプロセッサのようなアプリケーションソフトウェアであり、操作者からキーボード1105やマウス1106を用いて指示された内容に従って文書の作成・印刷などを行う。4はプリンタドライバであり、アプリケーション3が発行した印刷指令をOS2を介して受け取り、該印刷指令をプリンタ1111が解釈可能なプリンタコマンドに変換する。5はポートドライバであり、プリンタドライバ4が変換したプリンタコマンドをOS2を介して受け取り、不図示のパラレルポートを介してプリンタ1111に送信する。そしてプリンタ1111は、ポートドライバ5から受信したプリンタコマンドに従って印刷を行う。

【0021】図2はプリンタ1111の基本構成を示すブロック図である。図中、11はパラレルポートであり、ポートドライバ6から出力されたプリンタコマンドを受信する。12はFIFO（ファーストインファーストアウト）メモリであり、パラレルポート11が受信したプリンタコマンドに含まれる符号化データ（詳細は後述）を記憶し、記憶したデータを先入れ先出しの順に復号回路13に出力する。復号回路13は、FIFOメモリ12に記憶された符号列データを復号し、復元した画像データをプリンタエンジン14に出力する。プリンタエンジン14はレーザビームプリンタエンジンであり、制御回路15の指示により、復号回路13が出力した画像データに従って印刷を行う。画像データはシアン、マゼンタ、黄、黒の各色ごとに面順次で出力される。15は制御回路であり、例えば1チップCPUで構成され、パラレルポート11、FIFOメモリ12、復号回路13、およびプリンタエンジン14の制御を行う。

【0022】以下、印刷を行う際に、図1、2に示した各部の動作について以下、説明する。

【0023】操作者がキーボード1105やマウス1106を用いてアプリケーション3を操作して印刷データ

を生成し、更に、生成した印刷データを印刷する指示を入力すると、アプリケーション3からOS2を介してプリンタドライバ4に印刷指令が渡される。プリンタドライバ4はアプリケーション3から発行された印刷指令に基づき、赤、緑、青の3色からなるRGB画像データを作成し、次いでRGB画像データを、シアン、マゼンタ、黄、黒の4色からなるCMYK画像データに変換する。そして、プリンタドライバ4は、後述する符号化手順に基づき、作成した画像データから符号化データを生成し、用紙サイズ、画像データ（ビットマップデータ）のラインの長さとライン数などを指定する印刷制御コマンドとともにに出力する。ポートドライバ5は、プリンタドライバ4が作成した、印刷制御コマンドおよび符号化データからなる一連のプリンタコマンドを、プリンタ111に送信する。

【0024】制御回路15はパラレルポート11を経由してプリンタコマンドを受信する。受信したプリンタコマンドのうち印刷制御コマンドは、印刷制御のために制御回路15の内部に保持される。また、受信したプリンタコマンドのうち符号化データは、FIFOメモリ12に格納される。その後、ページ終了コマンドの受信などにより、1ページを構成するプリンタコマンドの受信が完了したことを制御回路15が検出したときに、プリンタエンジン14に印刷の開始を指示する。印刷の開始が指示されると、プリンタエンジン14は不図示の給紙カセットから用紙を給紙し、用紙が所定の位置に到達したときに、復号回路13に画像データの出力を要求する。復号回路13は、あらかじめFIFOメモリ12から符号化データを読み出し、復号することで復元される画像データを内部のバッファに保持しておき、プリンタエンジン14から画像データの出力を要求されたときに、内部のバッファに保持していた画像データを出力する。復号回路13は、画像データを保持していたバッファに空きができるときに、FIFOメモリ12から引き続く符号化データを読み出し、復号して内部のバッファに保持する。このようにして符号化データは順次復号されて画像データとして出力され、1ページの画像データの出力が全て終了すると、印刷が完了する。

【0025】次に、図3および図4に示すテーブルを参考し、図1に示したプリンタドライバ4が生成する符号について説明する。

【0026】図3は、図1に示したプリンタドライバ4が生成する符号化データに含まれる各符号の一例を説明するテーブルである。なお、本実施形態で説明する各符号はビット単位で可変長であり、例えば2ビットから18ビットまでのビット列で表現される。各符号はハフマン符号と同様に、先頭から順に調べることにより符号を識別することができるよう構成されている。

【0027】図3に示すように、符号のビット列が「1」で開始する場合はCOPY UPコマンドであ

る。このコマンドは、あらかじめ決定された所定ライン上の位置から、<バイト数>が示す長さのバイト列を複写することを指示する。

【0028】また、符号のビット列が「01」で開始する場合は、CACHEコマンドである。このコマンドはキャッシュバッファ（本実施形態ではCPU1101内に設けるが、これに限定されるものではなく、CPU1101の外部に設けても良い）内で<3ビットデータ>が示す位置の1バイトのデータを指定する。

【0029】また、符号のビット列が「001」で開始する場合はRAWコマンドである。このコマンドは<8ビットデータ>の値を持つ1バイトのデータを指定する。

【0030】また、符号のビット列が「0001」で開始する場合はCOPY LEFTコマンドである。このコマンドは、あらかじめ決定された所定バイト左の位置から、<バイト数>が示す長さのバイト列を複写することを指示する。

【0031】また、符号のビット列が「0000」の場合はEOPコマンドであり、符号化データの終了を指示する。

【0032】図4は上述のCOPY UPコマンド、COPY LEFTコマンドに後続する、長さを示す符号(<バイト数>)の一例を示すテーブルである。図4に示すように、符号のビット列が「1」である場合は1バイトの長さを指示する。

【0033】また、符号のビット列が「01」で開始する場合は1ビットのデータ（0もしくは1の値を取る）が後続し、後続する1ビットのデータが0の場合はコマンド符号としては「010」となり、意味としては2バイトを意味することになる。一方、後続する1ビットのデータが1の場合はコマンド符号としては「011」となり、意味としては3バイトを意味することになる。つまり、符号のビット列が「01」で開始する場合は、2バイト、もしくは3バイトの長さを意味することにある。

【0034】また符号のビット列が「001」で開始する場合は2ビットのデータが後続するので4バイトから7バイトまでの長さを意味し、符号のビット列が「0001」で開始する場合は3ビットのデータが後続するので8バイトから15バイトまでの長さを意味し、符号のビット列が「000001」で開始する場合は4ビットのデータが後続するので16バイトから31バイトまでの長さを意味し、符号のビット列が「0000001」で開始する場合は5ビットのデータが後続するので32バイトから63バイトまでの長さを意味し、符号のビット列が「00000001」で開始する場合は6ビットのデータが後続するので64バイトから127バイトまでの長さを意味し、符号のビット列が「00000000」で開始する場合は7ビットのデータが後続するので128バ

イトから255バイトまでの長さを意味する。

【0035】なお、これらの符号はあらかじめ多数の画像データで各コマンドの出現頻度を求めておき、ハフマン符号と同様に、出現頻度の高いコマンドに短い符号を、出現頻度の低いコマンドに比較的長い符号を割り当てるにより圧縮率を高くすることができる。

【0036】次に図5を参照し、RAWコマンドおよびCACHEコマンドの動作を説明するとともに、キャッシュバッファに格納されるデータについて説明する。図5は画像データと、この画像データがRAWコマンド及びCACHEコマンドによりキャッシュバッファに格納される方法を説明する図である。

【0037】同図に示すように、10バイトの画像データ00, 01, 02, 03, 04, 05, 06, 07, 08, 09が左から並んでいる。キャッシュバッファは8バイトの容量があり、最初は空である。

【0038】まず最初の画像データ00は符号0010000000すなわちRAW00コマンドに符号化することができ、このコマンドを符号化あるいは復号した結果、空であったキャッシュバッファの先頭にデータ00が格納される。

【0039】次の画像データ01は符号0010000001すなわちRAW01コマンドに符号化することができ、このコマンドを符号化あるいは復号した結果、キャッシュバッファの先頭にデータ01が格納され、先頭に格納されていたデータ00は次の位置に移動する。このようにして画像データ07までの8バイトのデータが符号化あるいは復号されるとキャッシュバッファは埋め尽くされる。次の画像データ08は符号001000100000すなわちRAW08コマンドに符号化することができ、このコマンドを符号化あるいは復号した結果、キャッシュバッファの先頭にデータ08が格納され、すでに格納されていたデータ07からデータ01までの7バイトのデータは順次次の位置に移動し、キャッシュバッファの最後に格納されていたデータ00は失われる。

【0040】次の画像データ06は同一のデータがキャッシュバッファの位置2に格納されているため、符号01010すなわちCACHE2コマンドに符号化することができ、このコマンドを符号化あるいは復号した結果、位置2に格納されていたデータ06はキャッシュバッファの先頭に移動し、位置2より前に格納されていたデータ08およびデータ07は順次次の位置に移動する。位置2より後に格納されていたデータは変更されない。

【0041】なお、COPY UPコマンドまたはCOPY LEFTコマンドを符号化あるいは復号した場合は、キャッシュバッファに格納されているデータは変更されない。

【0042】次に図6を参照し、COPY UPコマン

ドおよびCOPY LEFTコマンドの動作について説明する。図6は、画像データと、この画像データをCOPY UPコマンド及びCOPY LEFTコマンドに符号化する方法を説明する図である。同図に示すように、画像データを構成する画素列において、最下行には10バイトの画像データ01, 23, 45, 67, 89, AB, 89, AB, 89, ABが左から並んでおり、その4行上には画像データ01, 23, 45, 67, 89, AB, 00, 00, 00, 00が左から並んでいる。現在符号化あるいは復号しようとしている画像データは最下行であり、またCOPY UPコマンドは4行上を、COPY LEFTコマンドは2バイト左を参照するようあらかじめ設定されているものとする。

【0043】なお、上記画像データがディザマトリックスを用いてディザ処理を施された画像である場合、隣接するデータには異なった処理が行われるため、隣接したデータよりもむしろディザマトリックスの周期だけ離れた位置のデータのほうが相関が高くなるという傾向がある。このためディザ処理を施された画像の場合、COPY UPコマンドまたはCOPY LEFTコマンドが参照する位置は、注目位置との相関が最も高くなるように定められる。すなわち、符号化しようとしている画像データを生成したときに使用されたディザマトリックスの周期に応じて定められる。

【0044】ここで最下行の画像データのうち、先頭の6バイト01, 23, 45, 67, 89, ABは、4行上の先頭6バイトと同一の並びであるので、符号100110すなわちCOPY UP6コマンドに符号化することができる。また次の4バイト89, AB, 89, ABは同じ行の2バイト左から始まる4バイトと同一の並びであるので、符号000100100すなわちCOPY LEFT4コマンドに符号化することができる。

【0045】次に図7に示すフローチャートを参照し、プリンタドライバ4の処理の詳細を説明する。図7はプリンタドライバ4が行うメインの処理のフローチャートである。

【0046】プリンタドライバ4がオペレーティングシステム2からの指令を受信すると、指令の種類が描画指令であるか否かを判定する(ステップS5)。指令の種類が描画指令であった場合は、処理をステップS6に進め、描画処理を行う(ステップS6)。具体的には、オペレーティングシステム2を経由してアプリケーション3から送信された文字、図形またはビットマップ等を赤、緑、青の3色を用いた各色8ビットの画像に変換し、RAM1102に記録する。

【0047】一方、上記指令の種類が描画指令でなかった場合は、処理をステップS7に進め、更に、上記指令の種類がページ終了指令であるか否かを判定する(ステップS7)。指令の種類がページ終了指令であった場合

には、処理をステップS 8に進め、色変換処理を行う（ステップS 8）。具体的にはステップS 6においてRAM1102に記録された赤、緑、青の3色を用いた各色8ビットの画像を、所定のディザマトリックスを使用して、シアン、マゼンタ、黄、黒の4色からなる、例えば各色4ビットの画像に変換する。

【0048】次に、印刷条件指定コマンド、具体的には用紙サイズ、給紙カセット、解像度、階調数、1ラインのバイト数、1ページのライン数など印刷に必要な条件を指定するコマンド（印刷制御コマンド）を出力する（ステップS 9）。

【0049】そして次に、ステップS 10からステップS 14までの処理をシアン、マゼンタ、黄、黒の各色ごとに繰り返し、各色毎の印刷制御コマンドを出力する。各ステップにおける具体的な処理としては、まずステップS 8において使用したディザマトリックスに応じた圧縮パラメタ、すなわち符号化時に使用する、COPY UPコマンドおよびCOPY LEFTコマンドが参照する位置を出力する（ステップS 10）。次に後述する符号化手順に従い、画像データを符号化する（ステップS 11）。このときに、ステップS 10にて出力した圧縮パラメタが指定する、COPY UPコマンドおよびCOPY LEFTコマンドが参照する位置を用いて符号化を行う。次にステップS 11にて符号化された画像データのサイズおよびライン数を指定するヘッダを出力する（ステップS 12）。次にステップS 11にて符号化された画像データを出力する（ステップS 13）。次にシアン、マゼンタ、黄、黒の各プレーンの処理が全て終了したか否かを判定する（ステップS 14）。シアン、マゼンタ、黄、黒の各プレーンの処理が全て終了していない場合は処理をステップS 10に戻し、次のプレーンの処理を行う。一方、シアン、マゼンタ、黄、黒の各プレーンの処理が全て終了すると、処理をステップS 15に進め、ページの終了を指定するコマンドを出力する（ステップS 15）。

【0050】一方、ステップS 7における判断で、指令の種類がページ終了指令でなかった場合に、処理をステップS 16に進め、指令の種類に応じたその他の処理、例えばページ開始指令あるいはプリント能力問い合わせ指令等に対応する処理を行う（ステップS 16）。

【0051】次に、上記ステップS 11における符号化処理について、同処理の詳細を示すフローチャートを示す図8を参照して説明する。

【0052】まず、現在の位置、すなわち符号化する画素の位置を画像の左端上に設定する（ステップS 20）。次にCOPY UPの参照位置が有効であるか否かを判定する（ステップS 21）。具体的にはCOPY UPの参照位置が4行上である場合は、現在の位置が先頭行から4行以上である場合にCOPY UPの参照位置が有効であると判定する。COPY UPの参照位

置が有効である場合には処理をステップS 22に進め、現在の位置から始まるバイト列（画素データ列）と、COPY UPの参照位置から始まる画素データのバイト列を比較し、値が一致するバイト列の長さを求める（ステップS 22）。なおこの際に、行末に達した場合、及び長さが255バイトに達した場合には処理を打ち切るものとする。

【0053】次にステップS 22にて求めた長さが0であるか否かを判定する（ステップS 23）。0でない場合にはCOPY UPコマンドに符号化できると判断されるので処理をステップS 29に進め、COPY UPコマンド、すなわち符号1とそれに後続するバイト数（求めた長さ）を示す符号を出し、処理をステップS 32に進める。

【0054】一方、ステップS 21にてCOPY UPの参照位置が無効であると判定された場合、もしくはステップS 23にてステップS 22で求めた長さが0であると判定された場合は、処理をステップS 24に進め、COPY LEFTの参照位置が有効であるか判定する（ステップS 24）。

具体的にはCOPY LEFTの参照位置が2バイト左である場合は、現在の位置が左端から2バイト以上離れている場合にCOPY LEFTの参照位置が有効であると判定する。COPY LEFTの参照位置が有効である場合には処理をステップS 25に進め、現在の位置から始まるバイト列（画素データ列）と、COPY LEFTの参照位置から始まるバイト列を比較し、一致するバイト列の長さを求める（ステップS 25）。なおこの際に、行末に達した場合、及び長さが255バイトに達した場合には処理を打ち切るものとする。

【0055】次に、ステップS 25にて求めた長さが0であるか否かを判定する（ステップS 26）。0でない場合にはCOPY LEFTコマンドに符号化できると判断されるので処理をステップS 30に進め、COPY LEFTコマンド、すなわち符号0001とそれに後続するバイト数（求めた長さ）を示す符号を出し、処理をステップS 32に進める。

【0056】一方、ステップS 24にてCOPY LEFTの参照位置が無効であると判定された場合、もしくはステップS 26にてステップS 25で求めた長さが0であると判定された場合は、処理をステップS 27に進め、キャッシュバッファを検索し、現在の位置のバイト（画素データ）がキャッシュバッファに登録されているか否かを判定する（ステップS 27）。現在の位置のバイトがキャッシュバッファに登録されている場合、CACHEコマンドに符号化できると判断されるので、処理をステップS 31に進め、CACHEコマンド、すなわち符号01とそれに後続する、キャッシュがヒットしたキャッシュバッファの位置を示す3ビットデータを出力し（ステップS 31）、キャッシュバッファを更新する

(ステップS37)。具体的には現在の位置のバイトをキャッシュバッファの先頭に格納するとともに、キャッシュバッファの先頭から、キャッシュがヒットしたキャッシュバッファの位置の手前までのデータを順次キャッシュバッファの次の位置に移動する処理を行う。

【0057】一方、ステップS27にて現在の位置のバイトがキャッシュバッファに登録されていないと判定された場合には、処理をステップS28に進め、RAWコマンド、すなわち符号001とそれに後続する、現在の位置のバイトに等しい8ビットの生データを出力する

(ステップS28)。具体的には、現在の位置のバイトをキャッシュバッファの先頭に格納するとともに、キャッシュバッファの先頭からキャッシュバッファの最後の手前までのデータを順次キャッシュバッファの次の位置に移動する処理を行う。

【0058】次に、COPY UP、COPY LEFT、CACHE、あるいはRAWコマンドで処理されたバイト数だけ現在の位置を進める(ステップS32)。次に、全ての画像データを処理したか判定する(ステップS33)。全ての画像データを処理していない場合には処理をステップS21に戻し、上述の符号化処理を行う。また、全ての画像データを処理し終えた場合は、処理をステップS34に進め、EOBコマンド、すなわち符号0000を出力し(ステップS34)、更に、出力した符号の総ビット数が8の整数倍となるように(8の整数倍に達するまで)ビット0を出力する(ステップS35)。

【0059】次に、復号回路13について説明する。図9は復号回路13の基本構成を示すブロック図である。同図において入力バッファ21はFIFOメモリ12から読み出した符号化データを格納する。入力バッファ21は少なくとも4バイトのデータを格納することができ、バッファ21に空きが生じ、かつFIFOメモリ12にデータがある場合はFIFOメモリ12からデータを読み出して格納する。入力バッファ21はまた、ビットカウンタ23に保持された、処理済ビット数が8以上になった場合には、不要になった処理済データを破棄する。

【0060】セレクタ22は、例えば18組の8入力セレクタであり、入力バッファ21に格納された符号データを、ビットカウンタ23が示す処理済ビット数に従って選択することにより、コマンドデコード回路24が処理するために必要な、コマンドの開始位置合わせを行う。これは、入力バッファ21がバイト単位でデータを保持するのに対し、コマンドはビット単位の可変長データであるため、8箇所の開始位置があるために必要なものである。

【0061】ビットカウンタ23は、入力バッファ21に格納された符号データのうち処理済みのビット数を格納する。ビットカウンタ23はまた、コマンドデコード

回路24から出力された、コマンドのビット数をカウントし、ビットカウンタ23に格納された値をカウントした値に更新する。ビットカウンタ23はまた、入力バッファ21が処理済データを破棄した場合には、格納している値から破棄したビット数を減算する。ビットカウンタ23はまた、コマンドデコード回路24がEOBコマンドを復号したときに、コマンドデコード回路24からEOB信号を受信し、ビットカウンタの下位3ビットが全て0であれば何もせず、そうでなければ格納している値に8を加算するとともに下位3ビットをクリアする処理を行う。

【0062】コマンドデコード回路24は、例えばリードオンリーメモリ、あるいはワイヤードロジックにより構成され、セレクタ22によって位置合わせが行われた、入力バッファ21に格納された符号データを復号し、復号したコマンドに従ってキャッシュ出力回路25、上コピー出力回路26、左コピー出力回路27、生データ出力回路28、およびビットカウンタ23に、前述の、あるいは後述する各種の信号を出力する。

【0063】キャッシュ出力回路25は、コマンドデコード回路24がCACHEコマンドを復号したときにキャッシュバッファの位置を示す3ビットのデータを受信し、キャッシュバッファ36から該位置のデータを読み取って出力する。上コピー出力回路26は、コマンドデコード回路24がCOPY UPコマンドを復号したときにコピーすべきバイト数を受信し、受信したバイト数に応じてラインバッファ35からデータを読み取り、読み取ったデータを出力することを繰り返す。左コピー出力回路27は、コマンドデコード回路24がCOPY LEFTコマンドを復号したときにコピーすべきバイト数を受信し、受信したバイト数に応じて可変段シフトレジスタ39からデータを読み取り、読み取ったデータを出力することを繰り返す。生データ出力回路28は、コマンドデコード回路24がRAWコマンドを復号したときに生データを示す8ビットのデータを受信し、受信したデータを出力する。

【0064】ライン長レジスタ29は、受信した印刷制御コマンドに含まれる1ラインのバイト数を保持する。

列カウンタ30は、ラインバッファ35の現在の列アドレスを保持し、ラインバッファ35に書き込みが行われる毎にカウントアップし、またその結果、ライン長レジスタ29が保持する1ラインのバイト数に達した場合には0に復帰する。行数レジスタ31は、受信した圧縮パラメタに含まれるCOPY UPコマンドが参照する位置を示すライン数を保持する。行カウンタ32は、ラインバッファ35の現在の行アドレスを保持し、列カウンタ30がライン長レジスタ29が保持する1ラインのバイト数に達して0に復帰する毎にカウントアップする。なお、行数レジスタ31が保持するライン数に達した場合には0に復帰する。行カウンタ32はまた、行数レジ

スタ31にライン数が outputされたときには0に復帰する。

【0065】乗算回路33は、行カウンタ32が保持する現在の行アドレスと、ライン長レジスタ29が保持する1ラインのバイト数との積を計算することにより、ラインバッファ35の現在の行の先頭アドレスを出力する。加算回路34は、乗算回路33が出力する現在の行の先頭アドレスと、列カウンタ30が保持する現在の列アドレスとの和を計算することにより、ラインバッファ35の現在のアドレスを出力する。

【0066】ラインバッファ35は、複数のラインの復号データを保持し、加算回路34が出力するアドレスに従って復号データの入力または出力をを行う。ラインバッファ35は、行数レジスタ31が保持するライン数によって指定されるサイズのリングメモリを構成する。

【0067】キャッシュバッファ36は、キャッシュ出力回路25または生データ出力回路28が画像データを復号したときにこれを記憶し、またキャッシュ出力回路25が参照するデータを出力する。段数レジスタ38は、COPY LEFTコマンドが参照する位置を示すバイトオフセット値を保持する。可変段シフトレジスタ39は、シフトレジスタとセレクタによって構成され、段数レジスタ38が保持するバイトオフセット値に等しい段数のシフトレジスタを構成し、出力された復号データを、段数レジスタ38が保持するバイトオフセット値で指定された値より1だけ小さい回数遅延したデータを出力する。

【0068】以上の構成において、コマンドデコード回路24がCOPY UPコマンドをデコードすると、それに引き続く後続のバイト数をデコードし、上コピー出力回路26にデコードしたバイト数を出力する。そしてラインバッファ35において、現在の行よりも行数レジスタ31に格納されたライン数上で、且つデコードされたバイト数分の復号データ群がラインバッファ35から上コピー出力回路26に読み出され、上コピー出力回路26は読み出された復号データ群を加算回路34が出力するアドレス（言い換えれば、現在の位置）から順に書き込む。一方、読み出された復号データ群は可変段シフトレジスタ39に入力される（以下、先頭から）。そして可変段シフトレジスタ39に既に保持されていたデータは一段ずつシフトされ、可変段シフトレジスタ39の最終段からは、現在の位置の次の位置に対応する、COPY LEFTコマンドが参照する位置の1バイトが出力される。次いで列カウンタ30はカウントアップされる。このようにして、指定されたバイト数の処理が終了するまで、復号データの出力が行われる。

【0069】一方、コマンドデコード回路24がCOPY LEFTコマンドをデコードすると、それに引き続く後続のバイト数をデコードし、左コピー出力回路27にデコードしたバイト数を出力する。そしてこのバイト

数分、可変段シフトレジスタ39から復号データが可変段シフトレジスタ39から読み取られ、左コピー出力回路27に入力される。左コピー出力回路27がこの復号データを出力すると、出力された復号データはラインバッファ35の現在の位置（加算回路34が出力するアドレス）に書き込まれるとともに、可変段シフトレジスタ39に入力される。また可変段シフトレジスタ39に保持されていたデータは一段ずつシフトされ、可変段シフトレジスタ39の最終段からは現在の位置の次の位置に対応するCOPY LEFTコマンドが参照する位置の1バイトが出力される。次いで列カウンタ30はカウントアップされる。このようにして、指定されたバイト数の処理が終了するまで、復号データの出力が行われる。

【0070】コマンドデコード回路24がRAWコマンドをデコードすると、引き続く後続の8ビットの生データを生データ出力回路28に出力する。生データ出力回路28がこのデータを出力すると、出力された復号データは、ラインバッファ35の現在の位置に書き込まれるとともに、可変段シフトレジスタ39に入力され、また可変段シフトレジスタ39に保持されていたデータは一段ずつシフトされ、可変段シフトレジスタ39の最終段からは、現在の位置の次の位置に対応する、COPY LEFTコマンドが参照する位置の1バイトが出力される。次いで列カウンタ30はカウントアップされる。また、出力された復号データはキャッシュバッファ36の先頭に記憶される。

【0071】コマンドデコード回路24がCACHEコマンドをデコードすると、引き続く後続の3ビットのキャッシュバッファ内の位置を示すデータをキャッシュ出力回路25に出力する。キャッシュ出力回路25は指定された位置の復号データをキャッシュバッファ36から読み取り、その復号データを出力する。出力された復号データはラインバッファ35の現在の位置に書き込まれるとともに、可変段シフトレジスタ39に入力される。また可変段シフトレジスタ39に保持されていたデータは一段ずつシフトされ、可変段シフトレジスタ39の最終段からは現在の位置の次の位置に対応するCOPY LEFTコマンドが参照する位置の1バイトが出力される。次いで列カウンタ30はカウントアップされる。出力された復号データはまた、キャッシュバッファ36の先頭に記憶される。

【0072】次に図10を参照し、キャッシュバッファ36の構成について説明する。図10はキャッシュバッファ36の詳細な構成を示すブロック図である。

【0073】同図において51は第1のシフトレジスタであり、8ビット8段のシフトレジスタによって構成される。第1のシフトレジスタ51は生データ出力回路28またはキャッシュ出力回路25が出力した1バイトの画像データを8つまで記憶することができる。第1のシフトレジスタ51を構成する各段のレジスタは制御回路

55からシフトパルスが入力された場合には前段のレジスタが記憶しているデータを記憶する。

【0074】52は第2のシフトレジスタであり、1ビット8段のシフトレジスタによって構成される。第2のシフトレジスタ52を構成する各段のレジスタは第1のシフトレジスタ51の対応するレジスタにデータが記憶されているか否かを示す1ビットの情報を保持する。第2のシフトレジスタ52を構成する各段のレジスタは、制御回路55からシフトパルスが入力された場合には前段のレジスタが記憶しているデータを記憶する。

【0075】53はセレクタであり、入力されたアドレス、すなわち参照すべきキャッシュバッファの位置にしたがって、第1のシフトレジスタ51を構成する各段のレジスタのうちの1つが記憶しているデータを選択して出力する。

【0076】54はデコーダであり、キャッシュバッファからの読み取りを指示する読取信号が入力されたときに、入力されたアドレス、すなわち参照すべきキャッシュバッファの位置にしたがって、第2のシフトレジスタ52を構成する各段のレジスタのうちの1つをクリアする信号を出力する。

【0077】55は制御回路であり、キャッシュバッファへの書き込みを指示する書込信号が入力されたときに、第2のシフトレジスタ52に保持されている第1のシフトレジスタ51の対応するレジスタにデータが記憶されているか否かを示す情報にしたがって第1のシフトレジスタ51および第2のシフトレジスタ52のシフトが必要な各段に対してシフトパルスを出力する。シフトパルスはそれ以前の段のいずれかにデータが記憶されていない段がある場合には出力されず、そうでない場合には出力される。

【0078】生データ出力回路28が復号データを出力すると出力された復号データは第1のシフトレジスタ51に入力されるとともに、制御回路55に書込信号が入力される。データが記憶されている段は連続しているので、制御回路55は先頭段および前段にデータが記憶されている段にシフトパルスを出し、第1のシフトレジスタ51の先頭段には入力された復号データが、第2のシフトレジスタ52の先頭段にはデータが有効であることを示す値1がそれぞれ記憶されるとともに、前段にデータが記憶されている段には前段のデータが記憶される。

【0079】キャッシュ出力回路25がキャッシュバッファの位置を示す情報を受信すると、受信されたキャッシュバッファの位置を示す情報すなわちアドレスはセレクタ53およびエンコーダ54に入力されるとともに、読取信号がエンコーダ54に入力される。セレクタ53は入力されたアドレスにしたがって第1のシフトレジスタ51に記憶されているデータを選択して出力する。またエンコーダ54は、入力されたアドレスにしたがって

データが出力された段に対してクリア信号を出力する。この結果、第2のシフトレジスタ52のデータが出力された段のレジスタはクリアされる。

【0080】キャッシュ出力回路25はセレクタ53が出したデータを読み取り、それを復号データとして出力すると、出力された復号データは第1のシフトレジスタ51に入力されるとともに、制御回路55に書込信号が入力される。データが記憶されている段は第2のシフトレジスタ52のキャッシュバッファの読み取りが行われた段がクリアされたため不連続となっているので、制御回路55は先頭段からキャッシュバッファの読み取りが行われた段までの段にシフトパルスを出力するので、第1のシフトレジスタ51の先頭段には入力された復号データが、第2のシフトレジスタ52の先頭段にはデータが有効であることを示す値1がそれぞれ記憶されるとともに、第1のシフトレジスタ51および第2のシフトレジスタ52の第2段からキャッシュバッファの読み取りが行われた段までの段には、その前段のデータが記憶される。

【0081】このようにしてキャッシュバッファ36にはRAWコマンドまたはCACHEコマンドを復号したデータが新しい順に記憶されるように維持される。

【0082】以上の説明の通り、本実施形態の画像処理装置は、以前のデータ系列と一致する部分は一致する長さを符号化するので、画像データのように同一データの系列が並ぶことが多いデータを高い圧縮率で符号化することができる。

【0083】また、以前のデータ系列と一致しない場合にはキャッシュバッファを検索し、キャッシュがヒットした場合にはキャッシュバッファの位置を符号化するので、データ自体を符号化する場合に較べて短い符号に符号化することができ、以前のデータ系列と一致する部分が少ない場合でも極力圧縮率の低下を防ぐことができる。

【0084】また、データ系列が一致した長さを符号化した場合には、キャッシュバッファを更新しないので、長いデータ系列によりキャッシュバッファが書き換わることによる圧縮率の低下を防ぐことができる。

【0085】【第2の実施形態】第1の実施形態では復号処理をハードウェアで行っているが、これに限定されるものではなく、ソフトウェアで行っても良い。また、第1の実施形態では符号化(圧縮)処理をソフトウェアで行っているが、これに限定されるものではなく、専用のハードウェアを用いて行っても良い。又第1の実施形態では、符号化の単位データサイズは1バイトであったが、これに限定されるものではなく、例えば1画素あるいは2バイトであってもよい。

【0086】また第1の実施形態では、キャッシュバッファがフルのときに破棄するデータを選択する際に最も長い間参照されなかったデータを破棄する、いわゆるL

R U法を用いているがこれに限定されるものではなく、例えば擬似R U法、あるいは擬似ランダム法などを用いてもよい。

【0087】[他の実施形態] なお、本発明は、複数の機器（例えばホストコンピュータ、インターフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0088】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体（または記録媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム(OS)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0089】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0090】本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明した（図7、8の一部もしくは

\*は全部の）フローチャートに対応するプログラムコードが格納されることになる。

#### 【0091】

【発明の効果】以上詳細に説明したように、本発明では、以前の画像データ列と一致する場合にはその冗長性を利用して高い圧縮率を得るとともに、画像データ列が一致する部分が少ない場合においても、圧縮率の低下を極力抑えることができる。

#### 【図面の簡単な説明】

10 【図1】画像を印刷する際に用いるソフトウェア群とプリンタとの関係を示す概念図である。

【図2】プリンタ1111の基本構成を示すブロック図である。

【図3】図1に示したプリンタドライバ4が生成する符号化データに含まれる書く符号の一例を説明するテーブルを示す図である。

【図4】COPY UPコマンド、COPY LEFTコマンドに後続する、長さを示す符号（<バイト数>）の一例を示すテーブルを示す図である。

20 【図5】画像データと、この画像データがRAWコマンド及びCACHEコマンドによりキャッシュバッファに格納される方法を説明する図である。

【図6】画像データと、この画像データをCOPY UPコマンド及びCOPY LEFTコマンドに符号化する方法を説明する図である。

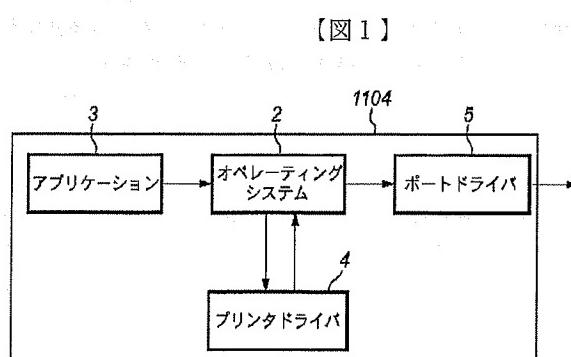
【図7】プリンタドライバ4が行うメインの処理のフローチャートである。

【図8】ステップS11における符号化処理の詳細を示すフローチャートである。

30 【図9】復号回路13の基本構成を示すブロック図である。

【図10】キャッシュバッファ36の詳細な構成を示すブロック図である。

【図11】本発明の実施形態における画像処理装置における基本構成を示すブロック図である。



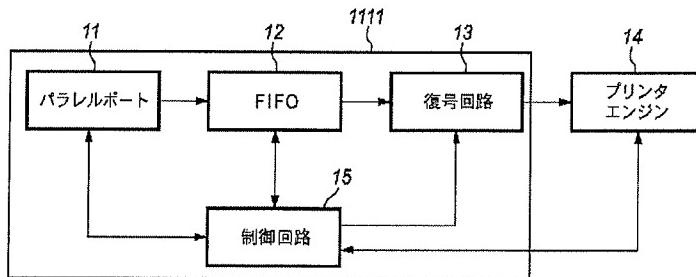
【図1】

画像データ									
01	23	45	67	89	AB	00	00	00	00
**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**
01	23	45	67	89	AB	89	AB	89	AB

符号	符号の意味
100110	COPY UP 6
000100100	COPY LEFT 4

【図6】

【図2】



【図3】

コマンド	コマンド符号	意味
COPY UP	1 <バイト数>	<バイト数>が示す長さの上からの複写
CACHE	01 <3ビットデータ>	キャッシュバッファの<3ビットデータ>が示す位置に格納されている1バイトのデータ
RAW	001 <8ビットデータ>	値が<8ビットデータ>である1バイトの生データ
COPY LEFT	0001 <バイト数>	<バイト数>が示す長さの左からの複写
EOB	0000	コマンドブロック終了

【図4】

コマンド	コマンド符号	意味
1	1	1バイト
2-3	01 <1ビットデータ>	2+<1ビットデータ> バイト
4-7	001 <2ビットデータ>	4+<2ビットデータ> バイト
8-15	0001 <3ビットデータ>	8+<3ビットデータ> バイト
16-31	00001 <4ビットデータ>	16+<4ビットデータ> バイト
32-63	000001 <5ビットデータ>	32+<5ビットデータ> バイト
64-127	0000001 <6ビットデータ>	64+<6ビットデータ> バイト
128-255	00000001 <7ビットデータ>	128+<7ビットデータ> バイト

【図5】

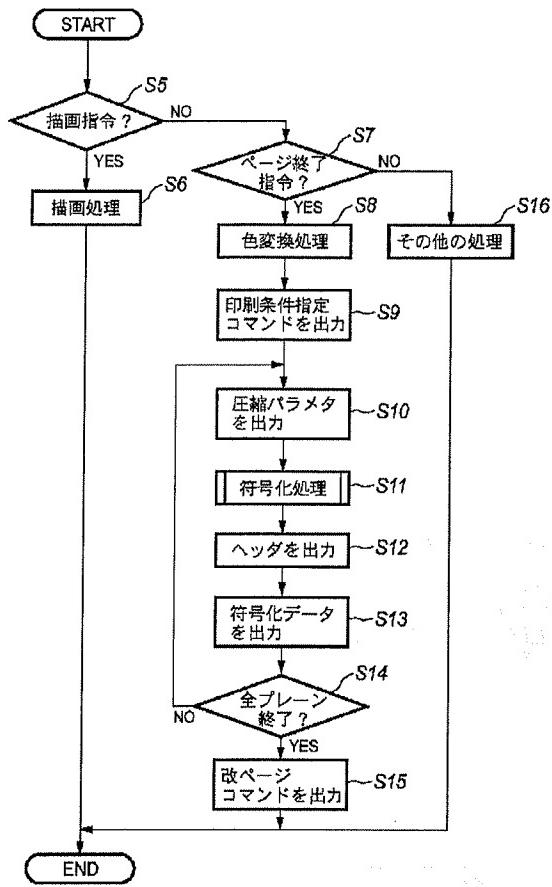
画像データ

00	01	02	03	04	05	06	07	08	06
----	----	----	----	----	----	----	----	----	----

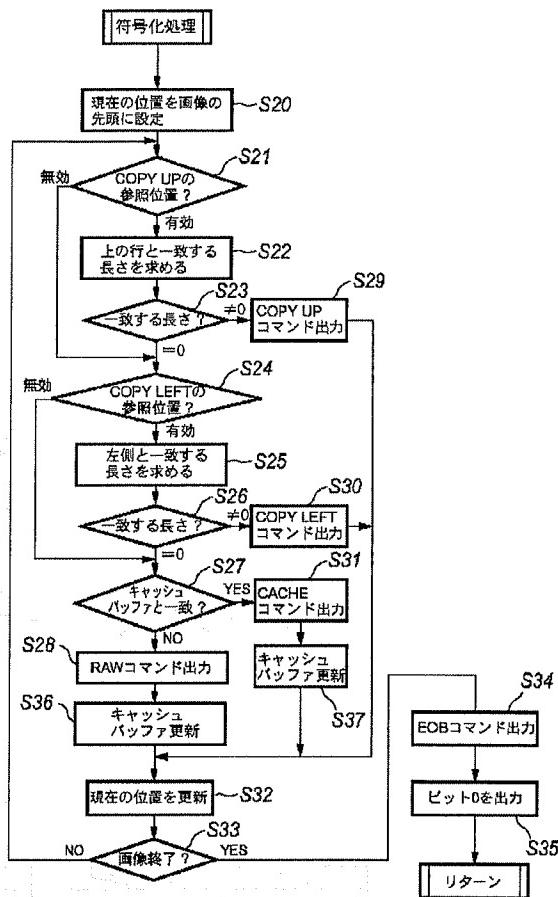
キャッシュバッファ

符号	符号の意味	0	1	2	3	4	5	6	7
001 00000000	RAW 00	00							
001 00000001	RAW 01	01	00						
001 00000010	RAW 02	02	01	00					
001 00000011	RAW 03	03	02	01	00				
001 00000100	RAW 04	04	03	02	01	00			
001 00000101	RAW 05	05	04	03	02	01	00		
001 00000110	RAW 06	06	05	04	03	02	01	00	
001 00000111	RAW 07	07	06	05	04	03	02	01	00
001 00001000	RAW 08	08	07	06	05	04	03	02	01
01 010	CACHE 2	06	08	07	06	04	03	02	01

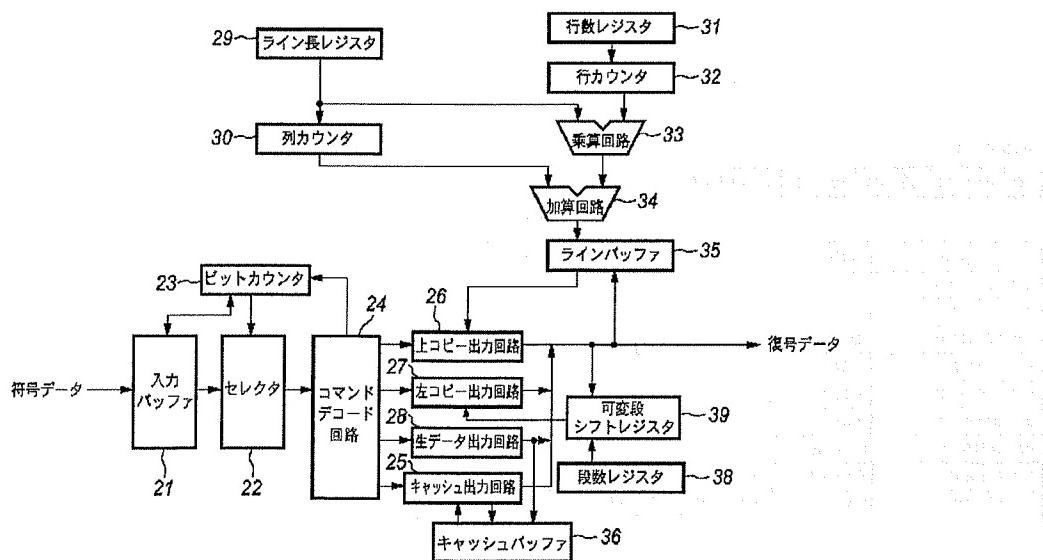
【図7】



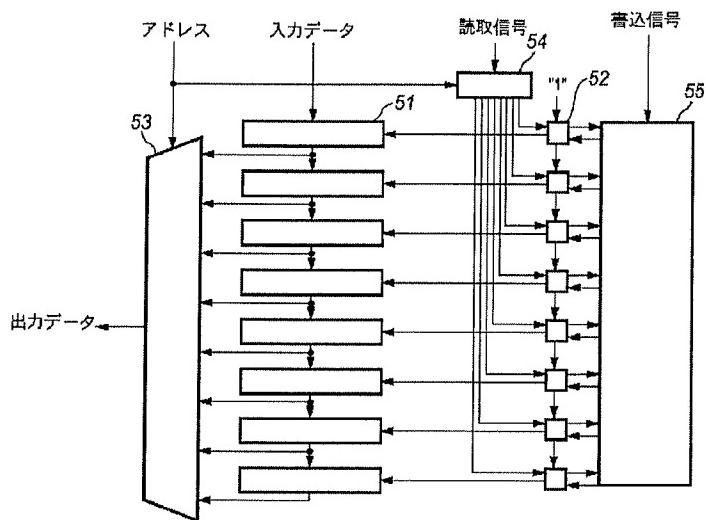
【図8】



【図9】



【図10】



【図11】

